**Module 6**

**Image manipulation in python**

Various modules are available for image manipulation in python.

**PIL(Python Image Library), opencv, simplecv, Imagemagick** are

some of them.

**Example**

from PIL import Image, ImageFilter

img = Image.open("dog.jpg")

img.rotate(45).show()

img\_sharp = img.filter( ImageFilter.SHARPEN )

img\_grey = img.convert(‘L’)

img\_sharp.show()

img\_grey.show()

**MySql and Python**

As you know Mysql is a popular open source database management system. We can access mysql database using python with the help of MySQLdb module.

**Example**

import MySQLdb

db = MySQLdb.connect(

host="localhost", # your host

user="root", # username

password="root", # password

db="dbname") # name of the database

# Create a Cursor object to execute queries.

cur = db.cursor()

# Select data from table using SQL query.

cur.execute("SELECT \* FROM examples")

# print the first and second columns

for row in cur.fetchall() :

print row[0], " ", row[1]

# Drop table if it already exist using execute() method.

cur.execute("DROP TABLE IF EXISTS EMPLOYEE")

# Create table as per requirement

sql = """CREATE TABLE EMPLOYEE (

FIRST\_NAME CHAR(20) NOT NULL,

LAST\_NAME CHAR(20),

AGE INT,

SEX CHAR(1),

INCOME FLOAT )"""

cursor.execute(sql)

sql = """INSERT INTO EMPLOYEE(FIRST\_NAME,

LAST\_NAME, AGE, SEX, INCOME)

VALUES ('Mac', 'Mohan', 20, 'M', 2000)"""

try:

# Execute the SQL command

cursor.execute(sql)

# Commit your changes in the database

db.commit()

except:

# Rollback in case there is any error

db.rollback()

db.close()

**PostgreSql and Python**

Postgresql is another popular open source DBMS. The postgresql database is accesible through python using psycopg2 module. As in

the case of mysql here also before accessing we have to establish a

connection to posgres.

**Example**

import psycopg2

conn = psycopg2.connect(database = "testdb", user = "postgres", password = "pass123",

host = 127.0.0.1",

port = "5432")

cur = conn.cursor()

cur.execute('''CREATE TABLE COMPANY (ID INT PRIMARY KEY NOT NULL, NAME TEXT NOT NULL,

AGE INT NOT NULL,

ADDRESS CHAR(50),

SALARY REAL);''')

print("Table created successfully")

cur.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)

'California', 20000.00 )")

cur.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)

'Texas', 15000.00 )")

cur.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)

'Norway', 20000.00 )")

cur.execute("INSERT INTO COMPANY (ID,NAME,AGE,ADDRESS,SALARY)

'Rich-Mond ', 65000.00 )")

conn.commit()

print("Records created successfully")

cur.execute("SELECT id, name, address, salary from COMPANY")

rows = cur.fetchall()

for row in rows:

print("ID = ", row[0])

print("NAME = ", row[1])

print("ADDRESS = ", row[2])

print("SALARY = ", row[3], "\n")

conn.commit()

conn.close()

VALUES (1, 'Paul', 32)

VALUES (2, 'Allen', 25)

VALUES (3, 'Teddy', 23)

VALUES (4, 'Mark', 25)

**Python objects and classes**

Python is an object-oriented language. All are(functions, data types, modules etc.) objects in python. In object-oriented programming objects are actually a collection of data and methods(functions) that act on this data.

For example string is an object in python and replace(),split(),join() are some methods of a string object.

A python class is a template or blue print for the object. Using this template(ie. class) we can create n number of objects. In Oops an object is also called as instance and creation of object is known as instantiation.

**Class definition in python**

A class with name Student can be created as follows.

class Student:

regno = 101

name = 'Madhav'

age = 15

def display(self):

print (self.name)

Here we defined a class Student with the following attributes:

data

regno, name and age

method

display() : function to display the name of student

The first argument of the function in class must be the object itself. This is conventionally called self,

We can create new object as follows

stu1 = Student()

stu1.display() will display the name of student.

If you want to change the name of student type

stu1.name = ‘Reshma’

In python there are some special functions with specific meaning.

These type of function begins with double underscore(\_\_) and are called constructors. \_\_init\_\_() is an example of constructor.

Class functions that begins with double underscore (\_\_) are called special functions as they have special meaning. This special function gets called whenever a new object of that class is instantiated. Normally it is used to initialize all variables.

class Student:

def \_\_init\_\_(self,r=0,n='madhu',a=0):

self.regno = r

self.age = a

self.name = n

def display(self):

print (self.name)

It is a bad to give access outside to the data fields used inside the class.

For example in the above example we can access the ‘regno’ variable from outside.

We can assaign value to regno as follows.

stud = Student()

stud.regno=123

To prevent this we can hide the data fields by declaring private data inside a class by giving double underscore(--) before the variable name.

For example we can hide the regno data field by putting underscore (\_\_regno)

We can also define a private method using two leading underscores.

class Student:

def \_\_init\_\_(self,r=0,n='madhu',a=0):

self.\_\_regno = r

self.\_\_age = a

self.\_\_name = n

def display(self):

print (self.\_\_name)

**Inheritance in python**

Deriving a new class from an existing class is known as inheritance.

Here the new derived class is derived/child class and the one from

which it inherits is called the base (or parent) class.

The syntax of inheritance is as follows

class BaseClass:

Body of base class

class DerivedClass(BaseClass):

Body of derived class

Example

class Person():

def \_\_init\_\_(self,name='madhav',

age=15,

[email='madhav@gmail.com](mailto:email%3D'madhu@gmail.com)',

phone=12345678):

self.name = name

self.age = age

self.email = email

self.phone = phone

def show(self):

print (self.name)

class Student(Person):

def \_\_init\_\_(self,regno=123,batch='A'):

self.regno = regno

self.batch = batch

def display(self):

print(self.name,self.regno,self.batch)

stud1=Student()

stud1.name='Karthik'

stud1.regno=5555stud1.batch='B'

stud1.show()

stud1.display()

In this example there are two classes Person and Student.

The class Person is base class and Student is derived from Person, hence it is a child class.

**Output**

Karthik

('Karthik', 5555, 'B')

The show function actually belongs to the class Person.

**Exercises**

1. What is garbage collection in python?

2. Explain with example the following python function

a. issubclass()

b. isinstance()

c. super()

3. Explain multiple inheritance and multilevel inheritance with proper examples in python.

4. Write a python program to create a new table with following structure.

Database name: xanthron

Table name : country

Structure

+--------------+---------------+------+-----+---------+-------+

| Field

| Type

| Null | Key | Default | Extra |

+--------------+---------------+------+-----+---------+-------+

| COUNTRY\_ID

| varchar(2)

| YES |

| NULL

|

|

| COUNTRY\_NAME | varchar(40)

| YES |

| NULL

|

|

| REGION\_ID

| decimal(10,0) | YES |

| NULL

|

|+--------------+---------------+------+-----+---------+-------+

5. Write a python program to insert records to the country table.

6. Write a program to read all the records from the country table and display it.

7. Write a class Company with following attributes.

Company name, address, total no of employees, Name of Director

methods to set these attributes to print the data

8. Write a python program to create a class named Triangle with method to calculate it’s area.

9. Write a python program to print all unique subsets of two numbers from a given set of distinct integer numbers using Oops techniques

10. Write a Python Oops program to find three numbers of a set of real numbers, sum of these three numbers should be equal to 25.